

We Claim:

5 1. A method of detecting one or more zombie global breakpoints for debugging computer software, said method including the steps of:

 checking a breakpoint data structure to determine if a breakpoint known to a debugging process is at an address where a breakpoint fired;

 if a known breakpoint cannot be determined at said address, verifying if a
10 breakpoint condition continues to exist at the address where the breakpoint fired; and
 if said breakpoint condition does not exist, identifying said breakpoint as a zombie breakpoint.

 2. The method according to claim 1, wherein said verifying step includes the step
15 of checking that a special breakpoint instruction exists at said address, being the exception location.

 3. The method according to claim 1, wherein said verifying step includes the step
20 of checking that an illegal breakpoint instruction exists at said address, being the exception location.

 4. The method according to claim 1, wherein said verifying step includes the step
25 of checking that said address, being the exception location, is present in a special debug register.

 5. The method according to claim 1, wherein physical settings for causing a breakpoint exception at a particular location are detectable from a breakpoint handler.

 6. The method according to claim 5, wherein breakpoint removal logic is
30 provided that lifts a physical breakpoint instruction from a breakpoint location before removing a breakpoint entry from said breakpoint data structure of said debugging process.

7. A computer-implemented apparatus for detecting one or more zombie global breakpoints for debugging computer software, said apparatus including:

a central processing unit for executing said computer software;

memory for storing at least a portion of said computer software;

5 means for checking a breakpoint data structure to determine if a breakpoint known to a debugging process is at an address where a breakpoint fired;

means for, if a known breakpoint cannot be determined at said address, verifying if a breakpoint condition continues to exist at the address where the breakpoint fired; and

10 means for, if said breakpoint condition does not exist, identifying said breakpoint as a zombie breakpoint.

8. The apparatus according to claim 7, wherein said verifying means includes means for checking that a special breakpoint instruction exists at said address, being the exception location.

15

9. The apparatus according to claim 7, wherein said verifying means includes means for checking that an illegal breakpoint instruction exists at said address, being the exception location.

20 10. The apparatus according to claim 7, wherein said verifying means includes means for checking that said address, being the exception location, is present in a special debug register.

25 11. The apparatus according to claim 7, wherein physical settings for causing a breakpoint exception at a particular location are detectable from a breakpoint handler.

30 12. The apparatus according to claim 11, wherein breakpoint removal logic is provided that lifts a physical breakpoint instruction from a breakpoint location before removing a breakpoint entry from said breakpoint data structure of said debugging process.

13. A computer program product having a computer readable medium having a computer program recorded therein for detecting one or more zombie global breakpoints

for debugging computer software, said computer program product including:

computer program code means for checking a breakpoint data structure to determine if a breakpoint known to a debugging process is at an address where a breakpoint fired;

computer program code means for, if a known breakpoint cannot be determined at
5 said address, verifying if a breakpoint condition continues to exist at the address where the breakpoint fired; and

computer program code means for, if said breakpoint condition does not exist, identifying said breakpoint as a zombie breakpoint.

10 14. The computer program product according to claim 13, wherein said computer program code means for verifying includes computer program code means for checking that a special breakpoint instruction exists at said address, being the exception location.

15 15. The computer program product according to claim 13, wherein said computer program code means for verifying includes computer program code means for checking that an illegal breakpoint instruction exists at said address, being the exception location.

20 16. The computer program product according to claim 13, wherein said computer program code means for verifying includes computer program code means for checking that said address, being the exception location, is present in a special debug register.

25 17. The computer program product according to claim 13, wherein physical settings for causing a breakpoint exception at a particular location are detectable from a breakpoint handler.

30 18. The computer program product according to claim 17, wherein breakpoint removal logic is provided that lifts a physical breakpoint instruction from a breakpoint location before removing a breakpoint entry from said breakpoint data structure of said debugging process.